Docket: 186232/US

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

| | | | |
|---|---|---|---|
| First Named Inventor: | Eng Boon Law | | |
| Appln. No.: | 10/523,128 | Confirmation No.: | 6469 |
| Filing Date: | January 21, 2005 | Examiner: | Lennox, Natalie |
| Title: | SYSTEM AND PROCESS FOR DEVELOPING A VOICE APPLICATION | Group Art Unit: | 2626 |

# AMENDMENT

Mail Stop Amendment
Commissioner for Patents
P. O. Box 1450
Alexandria, Virginia 22313-1450

Sir:

This Amendment is in response to the Office Action of May 1, 2008.

## AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Previously Presented) A process for developing a voice application, including:

generating graphical user interface components for defining execution paths of a voice application by arranging dialog elements in a tree structure, each path through said tree structure representing one of said execution paths, said dialog elements having user configurable properties and corresponding to respective predetermined sequences of VoiceXML elements;

receiving user input generated by user interaction with said graphical user interface components;

processing said user input to define a voice application by selecting dialog elements representing components of said voice application, configuring properties of the selected dialog elements, and defining execution paths of said voice application as respective sequences of at least a subset of the selected dialog elements; and

generating voice application code for said application, said application code representing each dialog element of said voice application as a sequence of VoiceXML elements including extended attributes to allow said tree structure of said application to be determined.

2-4. (Canceled)

5. (Previously Presented) A process as claimed in claim 1, wherein said extended attributes are qualified names of a qualified XML namespace.

6. (Previously Presented) A process as claimed in claim 1, wherein each dialog element of said application code includes a reference to the next of said dialog elements in an execution path of said application.

7. (Previously Presented) A process as claimed in claim 1, including processing said application code to generate a visual representation of said dialog elements and said execution paths.

8. (Previously Presented) A process as claimed in claim 1, wherein said step of generating application code includes generating extended VoiceXML code, prompt data, and grammar data for said application.

9. (Original) A process as claimed in claim 8, wherein said prompt data is represented as a grammar, and said process includes improving said grammar.

10. (Previously Presented) A process as claimed in claim 8, including generating at least one script for generating a prompt for said application on the basis of one or more parameters supplied to said script.

11. (Original) A process as claimed in claim 10, wherein said at least one script is generated on the basis of at least one script template and prompt data defined for said prompt by a user.

12. (Previously Presented) A process as claimed in claim 10, wherein said at least one script includes ECMAscript.

13. (Original) A process as claimed in claim 8, including generating VoiceXML code and IVR grammar data for execution of said application on an IVR system on the basis of said extended VoiceXML code, prompt data, and grammar data.

14. (Previously Presented) A system having components for executing the process of claim 1.

15. (Canceled)

16. (Previously Presented) A computer readable storage medium having stored thereon program instructions for executing the process of claim 1.

17. (Previously Presented) A system for use in developing a voice application, including:

a dialog element selector configured to define execution paths of said voice application by selecting dialog elements and adding said dialog elements to a tree structure, each path through said tree structure representing one of said execution paths, said dialog elements having user configurable properties and corresponding to respective predetermined sequences of VoiceXML elements;

means for receiving user input generated by user interaction with said dialog element selector;

means for processing said user input to define a voice application by selecting dialog elements representing components of said voice application, configuring properties of the selected dialog elements, and defining execution paths of said voice application as respective sequences of at least a subset of the selected dialog elements; and

a code generator for generating application code for said voice application, said application code representing each dialog element of said voice application as a sequence of VoiceXML elements including extended attributes to allow said tree structure of said voice application to be determined.

18. (Previously Presented) A system as claimed in claim 17, wherein said selector is adapted to process said application code to generate a graphical representation of said dialog elements and said execution paths of said application.

19. (Previously Presented) A system as claimed in claim 17, wherein said code generator generates extended VoiceXML code, prompt data, and grammar data for said application.

20. (Original) A system as claimed in claim 19, wherein said prompt data is represented as a grammar, and the system includes one or more modules for improving said grammar.

21. (Previously Presented) A system as claimed in claim 19, including a script generator for generating at least one script for generating a prompt for said application on the basis of one or more parameters supplied to said script.

22. (Previously Presented) A system as claimed in claim 21, wherein said script generator generates said at least one script on the basis of at least one script template and prompt data defined for said prompt by a user.

23. (Previously Presented) A system as claimed in claim 19, wherein said code generator generates VoiceXML code and IVR grammar data for execution of said application on an IVR system on the basis of said extended VoiceXML code, prompt data, and grammar data.

24. (Previously Presented) An extended VoiceXML file generated by the system of claim 17.

25. (Previously Presented) A graphical user interface for use in developing a voice application, said interface including graphical user interface components for defining execution paths of said application by arranging dialog elements in a tree structure, each path through said tree structure representing one of said execution paths, said dialog elements having user configurable properties and corresponding to respective predetermined sequences of VoiceXML elements, wherein said dialog elements include at least three of:

a start dialog component for defining the start of said application;

a variables component for use in defining variables for said application;

a menu component for defining a menu;

a menu choice component for defining a choice of said menu;

a decision component for defining a decision branching point;

a decision branch component for defining a test condition and an execution branch of said decision branching point;

a form component for defining a form to collect input from a caller;

a record component for recording audio;

a speaker component for playing prompts;

a local processing component for defining local processing;

a remote processing component for performing processing on a remote system;

a loop component for defining an execution loop;

a loop call component for calling said loop;

a loop next component for proceeding to the next cycle of said loop;

a loop break component for breaking out of said loop;

a subroutine component for defining a subroutine;

a subroutine call component for calling said subroutine;

a subroutine return component for returning from said subroutine;

a jump component for defining a non-sequential execution path to a dialog element;

a transfer component representing the transfer of a call to another number;

a hotwords component for defining a word or phrase and a non-sequential execution path to a dialog element to be followed upon receipt of said word or phrase; and

an end component for defining an end of said application.

26. (Previously Presented) A computer readable storage medium having stored thereon application code for a voice application, said application code including a plurality of dialog elements representing components of said voice application, each of said dialog elements being a sequence of VoiceXML elements including extended attributes to allow a tree structure of execution paths of said voice application to be determined, each path through said tree structure representing one of said execution paths.

## REMARKS

Claims 1, 5-14, and 16-26 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Voxeo Designer 2.0 (hereinafter "Voxeo") in view of Pfeiffer et al. (U.S. 2003/0055/651). In view of the following remarks, Applicants respectfully request formal allowance.

Voxeo discloses a visual phone markup design tool. Voxeo, 3rd para. Any CallXML or VoiceXML application may be opened in the Designer tool, updated graphically, and re-deployed for use. Voxeo, 3rd para. The workspace depicts a graphical representation of the application. Voxeo, 5th para. To add an element to the workspace, one simply clicks on an element in a toolbar. Voxeo, 5th para. The toolbar contains all of the CallXML or VoiceXML elements that are valid within the current editing context. Voxeo, 6th para. Properties for each element can be viewed and/or modified in the property editor. Voxeo, 7th para.

Pfeiffer et al. disclose a system, method, and computer program product for dynamically extending element types for a voice-based extensible mark-up language. Pfeiffer et al., Para. [0007]. A plurality of element types are registered with a VoiceXML interpreter. Pfeiffer et al., Para. [0007]. During use, the element may be received, and the extended type attribute associated with the element is identified. Pfeiffer et al., Para. [0010]. Thereafter, code corresponding to the registered type attribute may be accessed utilizing the VoiceXML interpreter. Pfeiffer et al., Para. [0010]. Such code extends the functionality of the VoiceXML. Pfeiffer et al., Para. [0010].

Claim 1 is patentably by calling for a process for developing a voice application as set forth therein, including "defining execution paths of a voice application by arranging dialog elements in a tree structure" (emphasis added).

Voxeo does not disclose, teach, nor suggest arranging dialog elements in a tree structure . It is understood by those skilled in the art that a tree structure is a structure that, in the context of application flow, involves a single initial flow (i.e., the tree trunk) that divides into at least two flows or branches. Depending on the application structure, each branch may then divide into at

least two sub-branches, and so on. The resulting structure looks like a tree (typically inverted, with the flow generally progressing from top to bottom), hence the name "tree structure."

The visual arrangement shown in the Voxeo document is not a tree structure. Each "flow" in the Voxeo document simply joins one element to another element. See, Voxeo Screen Shot. There is no illustrated possibility of one element branching directly to one of two or more other elements; every path originates from a "goto" statement without any alternative branch. See, Voxeo Screen Shot. Applicants' specification further points out this distinction:

> In contrast to arbitrary VoiceXML code whose execution can be completely non-sequential due to the presence of "GOTO" tags, a dialog generated by the system 100 has a tree structure, with each path through the tree representing a possible path of dialog execution. Para. [0048] (emphasis added).

As further evidence of Voxeo's lack of tree structure, apparent in the Screen Shot on page 1, the "ErrorBlock" element, for example, can be executed (or reached) by way of two alternative flows. Such an arrangement is not characteristic of a tree structure, wherein there is one and only one path from any point to any other point. See, e.g., Wikipedia, http://en.wikipedia.org/wiki/Tree_structure.

Additonally, neither Voxeo nor Pfeiffer et al., alone or in combination, disclose generating voice application code for said application, said application code representing each dialog element of said voice application as a sequence of VoiceXML elements including extended attributes to allow said tree structure of said application to be determined. The Examiner acknowledges that Voxeo does not specifically mention this limitation, but asserts that Pfeiffer et al., in paragraph [0010], teach this limitation.

Contrary to the assertion of the Examiner, Pfeiffer et al. in Paragraph [0010] merely refer to the use of extended type attributes *per se*. Pfeiffer et al. provide no disclosure or suggestion that the extended type attributes disclosed by Pfeiffer et al. could be used to determine a tree structure representing execution paths through dialog elements of a voice application as claimed and described in Applicants' specification:

> To facilitate the reverse translation from VoiceXML code to dialog, the dialog transformer 204 modifies the VoiceXML code

> by inserting additional attributes into various element tags, providing dialog element information that cannot be stored using the available VoiceXML tags. The resulting file 214 is effectively in an extended VoiceXML format. The additional attributes are stored in a separate, qualified XML namespace so that they do not interfere with the standard VoiceXML elements and attributes . . . This facilitates the parsing of extended VoiceXML files. Para. [0054].

Accordingly, Pfeiffer et al. do not teach or suggest generating voice application code for said application, said application code representing each dialog element of said voice application as a sequence of VoiceXML elements including extended attributes to allow said tree structure of said application to be determined as called for in Claim 1.

Additionally, Applicants maintain that Claim 1 is patentable by calling for "dialog elements having user configurable properties and corresponding to respective predetermined sequences of VoiceXML elements" and "generating application code for said application, said application code representing each dialog element of said application as a sequence of VoiceXML elements." and

In response to Applicants' remarks provided in their previous response, the Examiner disagreed with Applicants by asserting that Appendix A from Applicants' disclosure provides examples of dialog elements such as "Menu," "Record," "Speaker," and "Jump," among others which are the same as Voxeo's CallXML or VoiceXML elements present in the toolbar, such as "menu," "recordAudio," "playAudio," and "goto." However, the Examiner's assertion is conclusory and in direct conflict with Applicants' specification, which provides:

> Some dialog elements correspond to similar VoiceXML elements (e.g., a Menu dialog element corresponds to a VoiceXML <menu> element), while others map onto a complex sequence of VoiceXML elements (e.g., a Loop dialog element corresponds to multiple VoiceXML <form> elements, each form specifying the next form to execute in an iterative loop). However, even dialog elements that correspond to similar VoiceXML elements represent more functionality than the equivalent VoiceXML element. For example, a Menu dialog element allows prompts to be set by the user, and the Menu dialog element actually maps onto a block of VoiceXML code that contains a <menu> element with embedded

> <prompt>, <audio>, and other XML elements. Para. [0050] (emphasis added).

Therefore, contrary to the Examiner's assertion, dialog elements, as defined by Applicants' specification are not the same as Voxeo's CallXML or VoiceXML elements present in the toolbar. Rather, as stated above, the toolbar in Voxeo contains all of the CallXML or VoiceXML elements that are valid within the current editing context. Voxeo, 6th para. (emphasis added). In Voxeo, each of the basic graphical building blocks is a single CallXML or VoiceXML element. That is, a user in Voxeo graphically arranges individual CallXML or VoiceXML elements and configures them. Therefore, Voxeo does not disclose "dialog elements," as they are set forth in Claim 1. Applicants respectfully request that if the Examiner continues to assert that Voxeo discloses "representing each dialog element of said application as a sequence of VoiceXML elements," as recited in Applicants' Claim 1, the Examiner should provide evidence that Voxeo discloses more than basic VoiceXML and CallXML elements.

Claims 5-14 and 16 depend from Claim 1 and are patentable for the same reasons as Claim 1 and by reason of the additional limitations called for therein.

Claim 17 is patentable for reasons similar to Claim 1 by calling for a system for use in developing a voice application as set forth therein, including "defining execution paths of said application by selecting dialog elements and adding said dialog elements to a tree structure" (emphasis added). Claim 17 is additionally patentable by calling for "dialog elements having user configurable properties and corresponding to respective predetermined sequences of VoiceXML elements" and "generating application code for said application, said application code representing each dialog element of said application as a sequence of VoiceXML elements."

Claims 18-24 depend from Claim 17 and are patentable for the same reasons as Claim 17 and by reason of the additional limitations called for therein.

Claim 25 is patentable for reasons similar to Claim 1 by calling for a graphical user interface for use in developing a voice application as set forth therein, including "defining execution paths of said voice application by arranging dialog elements in a tree structure" (emphasis added). Claim 25 is additionally patentable by calling for "dialog elements having

user configurable properties and corresponding to respective predetermined sequences of VoiceXML elements."

Claim 26 depends from Claim 25 and is patentable for the same reasons as Claim 25 and by reason of the additional limitations called for therein.

In view of the foregoing, it is respectfully submitted that the claims of record are allowable and that the application should be passed to issue. Should the Examiner believe that the application is not in a condition for allowance and that a telephone interview would help further prosecution of this case, the Examiner is requested to contact Nathan Witzany at the phone number below.

This response is being submitted on or before September 2, 2008, with a request for an extension of time to that date, making this a timely response. The Commissioner is authorized to charge the fee for the extension of time to Deposit Account No. 04-1420. It is believed that no additional fees are due in connection with this filing. The Commissioner is also hereby authorized to charge any additional fees or credit any overpayments to Deposit Account No. 04-1420.

Respectfully submitted,

DORSEY & WHITNEY LLP
**Customer Number 75149**

Date: September 2, 2008

By: 
Nathan J. Witzany
Reg. No. 60,948
Telephone: (612) 492-6862